Extreme learning machines for predicting operation disruption events in railway systems

Olga Fink[a,*], Enrico Zio[b], Ulrich Weidmann[a]

[a] *Institute for Transport Planning and Systems, ETH Zurich, Zurich, Switzerland*
[b] *Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricité de France (EDF) at École Centrale Paris and SUPELEC, France; Department of Energy, Politecnico di Milano, Italy*

Abstract

European passenger rail systems are massively interconnected and operate with very high frequency. The impacts of component failures on these types of systems can significantly affect technical and operational reliability. Therefore, many advanced railway systems and components are equipped with monitoring and diagnostic tools to improve reliability and reduce maintenance expenditures.

Approaches to predict component failure and remaining useful life are usually based on continuously measured data. The use of event data is limited, especially for predicting failures in railway systems.

In this paper, we apply Extreme Learning Machines (ELM) to predict the occurrence of railway operation disruptions based on discrete-event data. ELM exhibit a good generalization ability, are computationally very efficient and do not require tuning of network parameters. For exemplification purposes, a case study with real data is considered concerning failures that cause undemanded service brake application of railway vehicles. While other machine learning techniques, such as multilayer perceptrons and feedforward neural networks with learning based on genetic algorithms, were not able to extract patterns in the diagnostic event data, the proposed approach was capable of predicting 98% of the operation disruption events correctly.

1. Introduction

Failures in European railway systems and networks have become more and more important because affecting wide areas of service and large numbers of users. To proactively react to this situation, railway infrastructure and rolling stock systems are being increasingly equipped with monitoring and diagnostic systems. These are intended to help operators identifying the locations of failures in short times and provide high reliability and availability of service. Additionally, they provide data that can be used to predict future failures.

To this aim, methods of fault detection and diagnosis, and remaining useful life prediction have been proposed in the literature (Marquez et al., 2007), (Chen et al., 2008), (Eker et al., 2011), (Yilboga et al., 2010). These methods typically used continuous data of monitored signals. Indeed, discrete event diagnostic data contain comparatively less information than continuously measured diagnostic signals: only predefined events with predefined parameters are logged when they occur. However, if diverse processes are monitored, the patterns of several occurring events can be combined to predict the patterns of the event of interest.

Generally, the approaches applied in the field of reliability prediction, fault diagnosis and prognosis are either model- and rule-based or they are data-based (Vachtsevanos, 2006). In the case of data-based approaches, the patterns derived from the example data patterns are used to directly transfer the patterns to the points of interest without deducing the functional relationship or a model valid for the entire possible input data space (Vapnik, 2006).

*Corresponding author
Email addresses: ofink@ethz.ch (Olga Fink), enrico.zio@ecp.fr (Enrico Zio), weidmann@ivt.baug.ethz.ch (Ulrich Weidmann)

Different data-based machine learning techniques have been applied in the field of reliability prediction, fault diagnosis and prognosis; among others different types of feedforward neural networks, such as Multilayer perceptrons (MLP) (Lolas and Olatunbosun, 2008), Radial-Basis-Function networks (RBF) (Zhang et al., 2011), and also Support Vector Machines (SVM) (Moura et al., 2011).

These methods suffer from drawbacks, such as local minima and time- and resource-consuming computations (Haykin, 2009). They also have several (hyper-)parameters to tune which is often done manually by experience or iteratively. There are also approaches of hyperparameter optimization (Bengio, 2000), by e.g. grid-search or genetic algorithms (Chatterjee and Bandopadhyay, 2012), but these approaches are also time- and resource-consuming.

Huang et al. (2004) introduced a new approach, so called Extreme Learning Machines (ELM), which overcome the shortcomings of several other machine learning techniques. They provide a fast and flexible learning algorithm, the parameters of which do not have to be determined manually by the user. Additionally, the method combines the advantages of several machine learning techniques, such as MLP, RBF and SVM. Besides its flexibility, the approach has also shown a good generalization ability on benchmark tasks, both in classification and in function regression (Huang et al., 2006b).

In this paper, we demonstrate the suitability of ELM algorithms to predict the occurrence of railway operation disruptions based on event-based diagnostic data. A case study is considered concerning failures on railway rolling stock systems that cause undemanded service brake application of railway vehicles.

2. Extreme Learning Machines

2.1. General concepts of Extreme Learning Machines

The ELM applied in this case study is a feedforward network with a single hidden layer and flexible processing units. ELM combine the strengths of several machine learning techniques, such as Support Vector Machines with kernels but also feedforward neural networks with different activation functions, e.g. sigmoidal and polynomial, and radial-basis functions (Huang et al., 2006b). The learning algorithm of ELM not only combines these activation functions within the hidden processing units, but also enhances the state-of-the-art approaches by speeding the learning process of the algorithms and by avoiding local minima, which gradient-based learning algorithms are prone to (Huang et al., 2006b).

Indeed, when gradient-based learning algorithms, such as backpropagation, are applied, the learning process is performed iteratively in several steps by propagating the error backwards through the network. The learning algorithm is therefore not efficient and leads often to locally optimal solutions (Haykin, 2009). Even though several modifications and improvements have been introduced to gradient-based learning algorithms (Haykin, 2009), they still have some limitations in terms of computational efficiency and generalization ability. Other machine learning techniques, such as SVM have overcome some of the limitations of neural networks with backpropagation learning algorithms (Haykin, 2009). However, they are still computationally intensive and require an iterative, often manual, parameter setting process.

Usually, the parameters of the applied machine learning algorithms are selected depending on the input and target example data. Determining the parameters of the algorithms is often performed iteratively, by genetic algorithms (Chatterjee and Bandopadhyay, 2012) or by hyperparameter optimization (Bengio, 2000). Contrary to the mentioned approaches, which require either a manual or a computationally very expensive parameter setting, all of the parameters of the ELM hidden nodes are not dependent on the target function or the training dataset (Huang et al., 2006b). Hidden nodes are chosen randomly by the algorithm and the output weights are determined analytically by determining the optimal combination of the output signals of the hidden layer (Huang et al., 2006b). Therefore, there is no need to set the parameters manually and to fine tune them iteratively. The parameters in the hidden layer can be independent of the training samples (Huang et al., 2011).

ELM have got some similarities to echo state networks (ESN). Similarly to ELM, main difference of ESN from other neural networks is that only the weights of the reservoir output signals are trained and are determined analytically. The weights of the connections within the reservoir are not trained but are generated randomly (Jaeger and Haas, 2004).

The major advantages of ELM are that they are computationally efficient, achieve good generalization ability, are not prone to local minima, and do not require manual parameter setting (Huang et al., 2006b).

2.2. Technical background of Extreme Learning Machines

The theoretical concepts and the computational algorithms were derived from (Huang et al., 2004), (Huang et al., 2006b), (Huang et al., 2006a), (Huang and Chen, 2007), (Huang and Chen, 2008), (Huang et al., 2011), (Huang et al., 2012) and are presented in the following.

The output function of the single hidden layer feedforward networks (SLFN) with L hidden nodes can be represented by the following equation:

$$\begin{aligned} f_L(\mathrm{x}) &= \sum_{i=1}^{L} \beta_i g_i(\mathbf{x}) \\ &= \sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}),\ \mathbf{x} \in \mathbf{R}^d,\ \beta_i \in \mathbf{R}^m \end{aligned} \tag{1}$$

where g_i denotes the output function $G(\mathbf{a}_i, b_i, \mathbf{x})$ of the ith hidden node, with $i = 1, \ldots, L$. For additive nodes with activation function g, g_i is defined as

$$g_i = G(\mathbf{a}_i, b_i, \mathbf{x}) = g(\mathbf{a}_i \cdot \mathbf{x} + b_i), \mathbf{a_i} \in \mathbf{R}^d, b_i \in \mathbf{R} \tag{2}$$

and for RBF nodes with activation function g, g_i is defined as

$$g_i = G(\mathbf{a}_i, b_i, \mathbf{x}) = g(b_i \|\mathbf{x} - \mathbf{a_i}\|), \mathbf{a_i} \in \mathbf{R}^d, b_i \in \mathbf{R}^+ \tag{3}$$

where $\mathbf{R}^+$ indicates the set of all positive real values.

For N arbitrary distinct samples $(\mathbf{x}_i, \mathbf{t}_i) \in \mathbf{R}^d \times \mathbf{R}^m$, standard SLFN with L hidden nodes are mathematically modelled as

$$\begin{aligned} \sum_{i=1}^{L} \beta_i g_i(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}_j) = \mathbf{o}_j, \\ j = 1, \ldots, N. \end{aligned} \tag{4}$$

The SLFN with L hidden nodes can approximate these N samples with zero error, $\sum_{j=1}^{L} \|\mathbf{o}_j - \mathbf{t}_j\| = 0$, i.e. there exist $(\mathbf{a}_i, b_i)$ and β_i such that

$$\sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}_j) = \mathbf{t}_j,\ j = 1, \ldots, N. \tag{5}$$

The above N equations can be rewritten as

$$\mathrm{H}\beta = \mathrm{T} \tag{6}$$

where

$$\begin{aligned} \mathrm{H} &= \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} \\ &= \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{x}_1) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{x}_N) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix}_{N \times L} \end{aligned} \tag{7}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \tag{8}$$

$$\text{and } \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N \times m} \tag{9}$$

$\mathbf{H}$ is the hidden layer output matrix of the neural network; the ith column of $\mathbf{H}$ is the ith hidden node output with respect to the inputs $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. $\mathbf{h}(\mathbf{x}) = G(\mathbf{a}_1, b_1, \mathbf{x}), \ldots, G(\mathbf{a}_L, b_L, \mathbf{x})$ is also referred to as hidden layer feature mapping with respect to the ith input $\mathbf{x}_i$: $\mathbf{h}(\mathbf{x_i})$.

The smallest least-squares solution of the above linear system is:

$$\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{10}$$

where $\mathbf{H}^{\dagger}$ is the *Moore-Penrose generalized inverse* of matrix $\mathbf{H}$. There are several methods that can be used to calculate the *Moore-Penrose generalized inverse* of a matrix, such as orthogonal projection methods, orthogonalization methods or iterative methods (Rao and Mitra, 1971).

The principles of applying ELM can be summarized as follows:

Given the training set $\aleph = \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbf{R}^d, \mathbf{t}_i \in \mathbf{R}^m, i = 1, \ldots, N\}$, hidden node output function $G(\mathbf{a}_i, b_i, \mathbf{x})$, and hidden node number L:

step 1	Generate randomly the values of the hidden node parameters $(\mathbf{a}_i, b_i), i = 1, \ldots, L$.
step 2	Calculate the hidden layer output matrix $\mathbf{H}$.
step 3	Calculate the output weight vector β : $\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{T}$

With the orthogonal projection method, $\mathbf{H}^{\dagger}$ can be reformulated as $\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ if $\mathbf{H}^T\mathbf{H}$ is nonsingular; or $\mathbf{H}^{\dagger} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}$ if $\mathbf{H}\mathbf{H}^T$ is nonsingular.

When ridge regression (Hoerl and Kennard, 1970) is applied, a positive value $1/\lambda$ is added to the diagonal of $\mathbf{H}^T\mathbf{H}$ or $\mathbf{H}\mathbf{H}^T$ in the calculation of the output weights β (Huang et al., 2012).

Including the ridge parameter, the corresponding output function of ELM becomes

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x})\beta = \mathbf{h}(\mathbf{x})\mathbf{H}^T\left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T}, \tag{11}$$

respectively:

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x})\beta = \mathbf{h}(\mathbf{x})\left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{H}^T\mathbf{T} \tag{12}$$

In the case of applying ridge regression, step 3 of the learning procedure corresponds to solving for β:

$$\hat{\beta} = \mathbf{H}^T\left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T} \tag{13}$$

or

$$\hat{\beta} = \left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{H}^T\mathbf{T} \tag{14}$$

3. Case study

3.1. Applied data

Real diagnostic discrete-event data from a European railway fleet consisting of 52 trains were used to demonstrate the applicability of the method to predicting the occurrence of failures that cause undemanded service brake application of railway vehicles, which cause disruption of operation. Some of the trains consist of 9 cars and the rest of 11. The available observation period was 313 days (approximately ten months). Data were collected automatically by event recorders, when a predefined diagnostic event occurred. The number and type of parameters recorded depend on the affected system. The parameters include speed, outside temperature, overhead line voltage etc. The recorded events are always assigned a time stamp, train location (i.e. train number, car number) and actual location via GPS.

It is assumed that given the size of the fleet, the dataset covers all relevant combinations of the different pertinent parameters and that the consequences are reflected in the occurrences of the diagnostic events. Hence, the data are considered sufficient to demonstrate the feasibility of the approach.

There are 255 distinct event codes for the brake system considered in this research. The diagnostic events fall into different categories:

– Driver action required – high priority;

– Driver action required – low priority;

– Driver information;

– Maintenance.

Depending on the category, the corresponding events will be communicated to the driver or only to the maintenance crew. High-priority diagnostic events are those that can potentially result in a delay-causing event. The high-priority event used for predictions in this study was the undemanded service brake application.

3.2. Data preprocessing and applied approach

To ensure that the information on the predicted railway operation disruption event can be effectively used to anticipate and prevent its occurrence, a time to react and anticipate failure is considered. The time to react is required for rescheduling, planning and preparation of maintenance. This time depends on the specific maintenance conditions and operational planning, i.e. timetable, vehicle scheduling and duty scheduling. For this case study the time to react was defined as seven days. Within this time interval no operational disturbances are supposed to occur to enable adaptations in operational planning.

In this research, a pattern of recorded parameters values was classified as belonging to class "DE" (impending operational **d**isruption **e**vent), if within the time period of seven days, starting from the end of the anticipation period, at least one operational disruption event would occur. If no operational disruption event occurred during the prediction period, the time pattern was classified as belonging to class "NE" (**n**o occurring **e**vents). Thus, it is a binary classification task. Contrary to multi-class approaches, binary classification tasks increase the selectivity of the algorithms (Duda et al., 2001).

The input data patterns represent for each of the 255 distinct events the time elapsed from the specific observation point of time to the last occurred event. This approach enables integrating information on the time series of the occurring distinct events in the input patterns. However, the information on the density of the occurring events, which is especially important if several events occur within a short period of time, is neglected by applying this approach.

In this case study, the observation time window was set to four weeks because this was considered sufficiently long for different diagnostic event patterns to evolve and given the amount of available data (ten months). In order to cover all possible combinations of diagnostic events and also to generate a sufficient number of input signals, the data patterns were generated by moving a four-week fixed time window over the 313-day study period, one day at a time. The consequence of this approach is that the time periods overlap and the data patterns can show similarities. However, these similarities will be observable not only within one class, but also between the classes.

The input dataset was preprocessed before presenting the data to the algorithm: the inputs were normalized to have zero mean and unit standard deviation.

In order to ensure a good learning capability of the algorithm, the input dataset was balanced in such a way that the dataset was composed of equal numbers of patterns from both classes. This approach is valid only if the selected data patterns from one class are representative for the entire class. An alternative approach to balance the dataset would be to draw additional data samples from the class underrepresented with replacement from the input dataset. This corresponds to the bootstrap method, which is often applied if there are not sufficient patterns from one class to achieve a good generalization ability of the algorithm (Efron and Tibshirani, 1993). However, the bootstrap approach does not increase the information content in the data patterns of the underrepresented class.

Subsequently, the data sequence is randomized in order to ensure that the generalization ability of the algorithm is not affected by the sequence of the presented patterns.

The parameters of the ELM do not have to be set and tuned manually, but are either set randomly or determined within the learning procedure.

After preprocessing the input data, they were presented to the ELM algorithm and the algorithm was trained.

Ridge regression with a regularization term of 0.01 (respectively $\lambda = 100$ was applied). In ridge regression, a regularization term is included in the minimization of residuals, to impose rigidity (Equation 13 or 14).

The process of the applied approach is demonstrated in Figure 1.

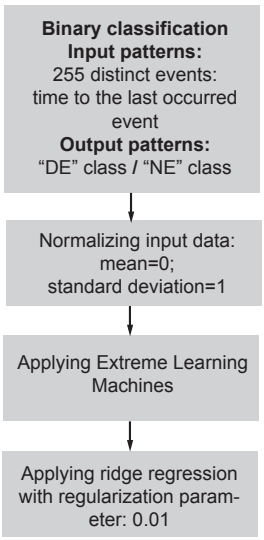

Figure 1: Applied approach

3.3. Validating the proposed approach on the case study

To validate the performance of the applied algorithm the holdout approach was applied. First, the available dataset was subdivided in two disjoint subsets. The first subset, referred to as training dataset, containing 90% of all the data patterns, was used to learn the underlying input-output relationships from the presented training patterns. After the training, the algorithm was tested on the remaining 10% of the input patterns (test dataset) different from those of training.

In total there were 3774 patterns in the dataset, equally divided between the ”DE“ class and ”NE” class. The training subset contained 3127 patterns; the testing subset contained 347 patterns.

In binary classification tasks (i.e. classification in two classes), positives are defined as the data samples from the class with the specified condition of interest while negatives are data samples from the other class (i.e. which do not have the specified condition). In this research, the positives were defined as data samples from ”DE“ class, leading to operational disruptions. For classification tasks, the correctly identified positive patterns are referred to as *true positives* (TP) and correctly identified negative patterns as *true negatives* (TN) (Han et al., 2011). The patterns, that are incorrectly classified by the classifier as negative patterns are referred to as *false negatives* (FN) and patterns incorrectly classified as positives as *false positives* (FP) (Han et al., 2011).

To estimate the performance of a classifier either the average performance for all presented classes is assessed, which can be defined by the misclassification rate, and/or the performance for a single class (how well the algorithm distinguished between the single classes) is assessed, which is defined by the sensitivity and specificity of a classifier.

The misclassification rate (Equation 15) can be defined as the ratio of the sum of all the incorrectly classified patterns to all patterns in the considered dataset (either training or testing dataset) (Han et al., 2011).

$$MR = \frac{FN + FP}{TP + FN + FP + TN} \tag{15}$$

To evaluate the ability of the algorithm to discriminate between the single classes, sensitivity and specificity parameters are assessed. Sensitivity measures the ability to identify the positives, while specificity measures the ability to identify the negatives. Sensitivity is also referred to as the *true positive rate* (TPR) (Equation 16 below) and specificity is referred to as *true negative rate* (TNR) (Equation 17 below) (Han et al., 2011). The higher the sensitivity and specificity, the better the performance. There is usually a trade-off between the two measures and the weights between them can be adjusted depending on whether the

costs of false positives (e.g. replacing parts that may not fail) are higher than those of false negatives (e.g. events that have not been detected by the algorithm and could cause severe service disruptions). In this study, the costs for false positives and negatives are considered equally important and therefore sensitivity and specificity were weighted equally.

$$TPR = \frac{TP}{TP + FN} \tag{16}$$

$$TNR = \frac{TN}{TN + FP} \tag{17}$$

For the applied case study, the misclassification rate for the training dataset was 1.85%, and 2.02% for the testing data. Thus, the algorithm was capable to generalize well on the testing dataset and showed a similar performance on the training and on the testing dataset.

The sensitivity of the proposed algorithm is 98.20%, which means that 98.20% of all the patterns from the "DE" class in the testing dataset were correctly discriminated from the "NE" patterns. The specificity of the applied classifier is 97.78%, meaning that 97.78% of all the patterns from the "NE" class in the testing dataset were classified correctly. The fact that both specificity and sensitivity are high and are in the same value range confirms that the algorithm is not biased towards either of the two classes and learned to discriminate well between both classes.

To compare the generalization ability of the algorithm to that of other machine learning techniques, the input patterns were also presented to a standard MLP and a feedforward neural network trained with genetic algorithms. However, even with a substantial number of neurons in the hidden layer and with several hidden layers, the networks were not able to discriminate the patterns of the two classes.

4. Discussion and conclusions

The study demonstrates the application of extreme learning machines to extract information on the condition of railway vehicle systems from diagnostic event data and to use the extracted patterns for predicting occurring operational disruption events. In the case study, the occurrence of undemanded service brake application of railway vehicles is predicted. The ELM shows a good generalization ability and leads to good results in terms of precision of classifying potentially occurring disruptions in railway operation. Additionally, ELM proves to be a very fast and efficient learning algorithm, especially compared to approaches with gradient based learning algorithms. Despite the randomly chosen input weights and the biases of the hidden layer, the algorithm shows a good generalization ability.

The misclassification rate achieved with the proposed approach is 2% and the algorithm is not biased towards one of the classes, showing similar values for sensitivity and specificity for both classes. The patterns that were not correctly classified by the algorithm could also be random events without a clear pattern leading to their occurrence. The misclassification of these patterns could have possibly not been caused by a lacking generalization ability of the algorithm for these patterns, but by the random character of the occurring events.

The results of the case study confirm that ELM is a powerful, very fast and efficient learning algorithm, especially compared to approaches with gradient-based learning algorithms. The algorithm can be applied in a very flexible way and shows several advantages over the state-of-the-art machine learning techniques, such as good generalization ability, computational efficiency and efficient parameter setting. The performance of the algorithm can be increased by applying ridge regression to find the optimal combination of the output signals.

One of the advantages of the ELM algorithm is that it can be applied to higher input dimensions without significantly affecting its speed and efficiency. Up to now, only events affecting the considered subsystems have been used as input for the classification task. However, the interaction of different subsystems and their mutual influence have not yet been considered within the prediction task. Additionally, occurrence of operational disrupting events caused by distinct systems have not been predicted in parallel. Integrating mutual influences and interaction of the distinct systems, as well as the joint prediction for several subsystems in parallel may be possible with the ELM approach, due to its efficient computations and good generalization, even with very high-dimensional input. Furthermore, additional input sources can be included in the consideration for increasing the significance of the results and without affecting computational speed or complicating the parameter setting process.

5. ACKNOWLEDGEMENTS

The authors would like to thank ALSTOM Transportation for providing the data for this research project.

The participation of Olga Fink to this research is partially supported by the Swiss National Science Foundation (SNF) under grant number 205121_147175.

The participation of Enrico Zio to this research is partially supported by the China NSFC under grant number 71231001.

6. References

Bengio, Y., 2000. Gradient-based optimization of hyperparameters. Neural Computation 12 (8), 1889–1900.

Chatterjee, S., Bandopadhyay, S., 2012. Reliability estimation using a genetic algorithm-based artificial neural network: An application to a load-haul-dump machine. Expert Systems with Applications 39 (12), 10943–10951.

Chen, J., Roberts, C., Weston, P., 2008. Fault detection and diagnosis for railway track circuits using neuro-fuzzy systems. Control Engineering Practice 16 (5), 585–596.

Duda, R. O., Hart, P. E., Stork, D. G., 2001. Pattern classification, 2nd Edition. John Wiley, New York.

Efron, B., Tibshirani, R., 1993. An introduction to the bootstrap. Chapman & Hall, New York [etc.].

Eker, O. F., Camci, F., Guclu, A., Yilboga, H., Sevkli, M., Baskan, S., 2011. A simple state-based prognostic model for railway turnout systems. Industrial Electronics, IEEE Transactions on 58 (5), 1718–1726.

Han, J., Kamber, M., Pei, J., 2011. Data mining concepts and techniques, 3rd Edition. Morgan Kaufmann, San Francisco, Calif.

Haykin, S. S., 2009. Neural networks and learning machines, 3rd Edition. Pearson Education, Upper Saddle River.

Hoerl, A. E., Kennard, R. W., 1970. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 12 (1), 55–67.

Huang, G.-B., Chen, L., 2007. Convex incremental extreme learning machine. Neurocomputing 70 (1618), 3056–3062.

Huang, G.-B., Chen, L., 2008. Enhanced random search based incremental extreme learning machine. Neurocomputing 71 (1618), 3460–3468.

Huang, G.-B., Chen, L., Siew, C.-K., 2006a. Universal approximation using incremental constructive feedforward networks with random hidden nodes. Neural Networks, IEEE Transactions on 17 (4), 879–892.

Huang, G.-B., Wang, D., Lan, Y., 2011. Extreme learning machines: a survey. International Journal of Machine Learning and Cybernetics 2 (2), 107–122.

Huang, G.-B., Zhou, H., Ding, X., Zhang, R., 2012. Extreme learning machine for regression and multiclass classification. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 42 (2), 513–529.

Huang, G.-B., Zhu, Q.-Y., Siew, C.-K., 2004. Extreme learning machine: a new learning scheme of feedforward neural networks. In: Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on. Vol. 2. pp. 985–990 vol.2.

Huang, G.-B., Zhu, Q.-Y., Siew, C.-K., 2006b. Extreme learning machine: Theory and applications. Neurocomputing 70 (1), 489–501.

Jaeger, H., Haas, H., 2004. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. Science 304 (5667), 78–80.

Lolas, S., Olatunbosun, O. A., 2008. Prediction of vehicle reliability performance using artificial neural networks. Expert Systems with Applications 34 (4), 2360–2369.

Marquez, F. P. G., Weston, P., Roberts, C., 2007. Failure analysis and diagnostics for railway trackside equipment. Engineering Failure Analysis 14 (8), 1411–1426.

Moura, M. d. C., Zio, E., Lins, I. D., Droguett, E., 2011. Failure and reliability prediction by support vector machines regression of time series data. Reliability Engineering & System Safety 96 (11), 1527–1534.

Rao, C. R., Mitra, S. K., 1971. Generalized inverse of matrices and its applications. Wiley, New York [etc.].

Vachtsevanos, G., 2006. Intelligent fault diagnosis and prognosis for engineering systems. Wiley, Hoboken, NJ.

Vapnik, V. N., 2006. Estimation of dependences based on empirical data reprint of 1982 ed. afterword of 2006, 2nd Edition. Springer New York, New York, NY.

Yilboga, H., Eker, O. F., Guclu, A., Camci, F., 2010. Failure prediction on railway turnouts using time delay neural networks. In: Computational Intelligence for Measurement Systems and Applications (CIMSA), 2010 IEEE International Conference on. pp. 134–137.

Zhang, K., Li, Y., Scarf, P., Ball, A., 2011. Feature selection for high-dimensional machinery fault diagnosis data using multiple models and radial basis function networks. Neurocomputing 74 (17), 2941–2952.